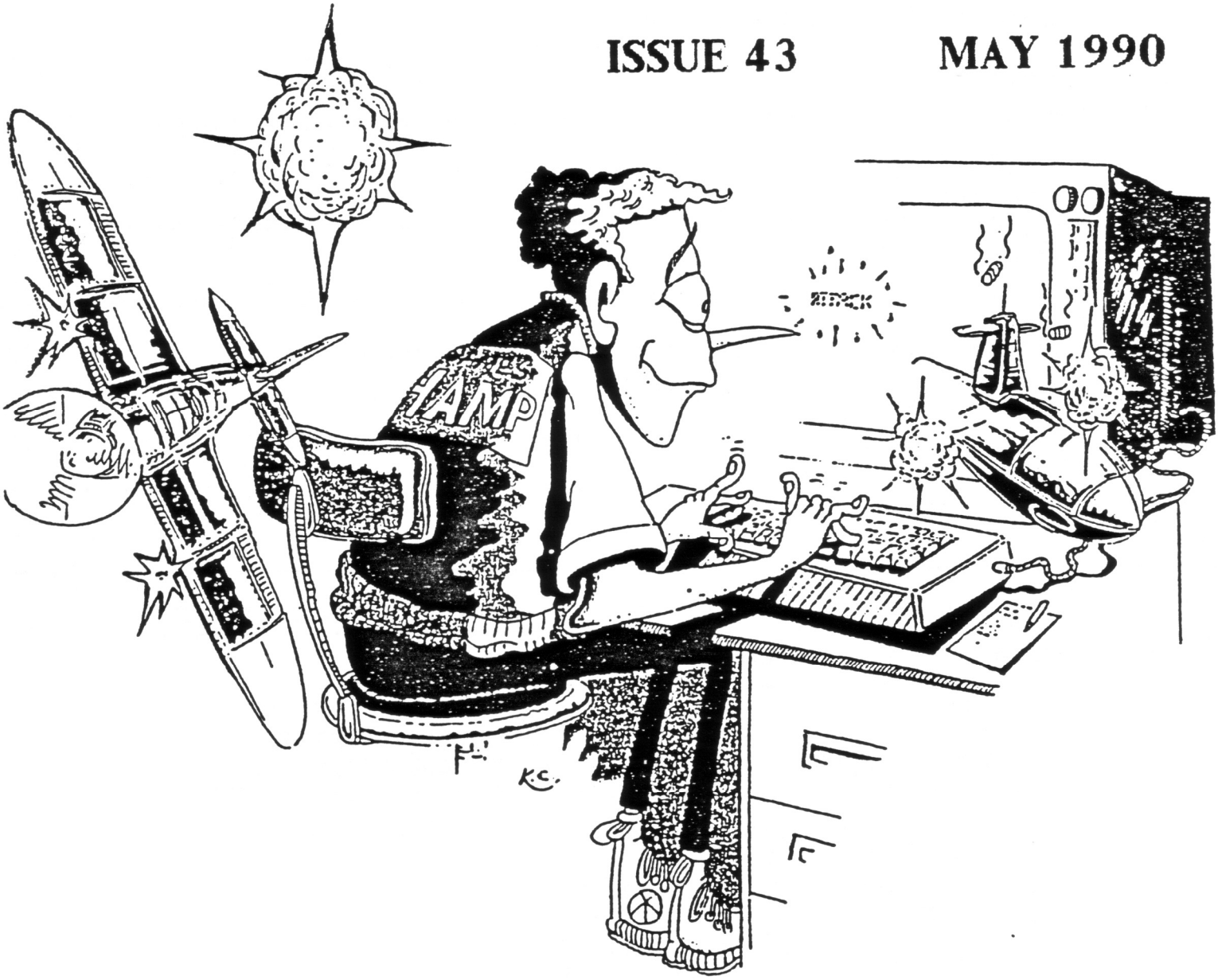
VZ LINK magazine

New Zealand VZ 300/200 Computer users club

ISSUE 43 MAY 1990



FROM ME TO YOU

Here we are again with another super issue of VZ LINK for your enjoyment.

With the nights becoming darker and colder, I hope you all get time to fire up the VZ and maybe write an item for your newsletter.

WORKSHOP/MEETING

I would like to thank all those that attended the meeting on the 7 April. We had a really enjoyable day. We got a chance to see some of the software which is available.

We also had a preview of some really exciting programs which will appear in this newsletter over the next couple of months.

SPECIAL THANKS

To Lou Ellmers who made the effort and travelled down from Whangarei. Also Martin and family who came down from Warkworth.

Also to Bob Kitch for suppling software to demonstrate at the meeting. I think I speak for everyone when I say that we look forward to meeting Bob at a meeting in the near future.

NEXT ISSUE

We have another very good program called Fisherman written by Ken Clarke.
A item on Battery back-up ram by Dave Boyce.
Error handling routines.
And we continue the Micro Magic series with a look at sorting, sound effects etc.

I will also start the VZ gamers highscore list in the June issue, so start now.
This list is open to all gamers and for any VZ game.

So until next issue happy VZing.

P.O BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USERS CLUB (1986)
NEW ZEALAND VZ 300/200 COMPUTER USERS CLUB (1989)

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT.
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT IT IS FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR ANY BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OR USE OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME IF YOU HAVE PROBLEMS.
PHONE 761-451 AFTER 6.00 PM.

Learning to program — the why, who, what which & how

By Gordon Findlay

Booting up

Each article in this series is a gentle introduction to some topic in the computing field. It is written for the beginner, so may appear very simple to the rest of you. If you find it too easy and not worth reading, congratulations — you are a beginner no more. Each issue will deal with a different topic, of general interest. Occasionally material may seem to repeat that which has already appeared in "Bits and Bytes". Remember, new readers are coming along all the time.

The title of the column comes from a computing term; "booting up" is whatever must be done to get a computer ready to run programs.

Home computers have already passed through three phases in their brief adolescence, each characterised by the sorts of things their owners have done with them.

The first phase was that of the builders. They either built a computer from scratch or a kit, or at least weren't averse to getting in with a soldering iron and using a junk teletype as a printer.

Next came the hackers. This has now become a pejorative term, relating to fools who break into communication and computer systems which don't belong to them. Until recently though, a hacker was a person who programmed — often in a way to get more out of a machine than could rightfully be expected, but always compulsively. These are the people we have to thank for much of the software we use now. They have written the books and magazine articles which we still use as sources, and as inspiration. A typical activity — using two years of spare time to find out how one version of BASIC worked.

Now we have the users. There are still, thank goodness, a lot of hobbyists who program, and program well. There are still just as many, or more, people willing to find out how to make the machine do odd things. But I suspect, nay fear, most buyers now have little intention of programming.

What has caused this turnaround? There are several factors. One, is of course, the increasing amount of good software available to buy. The increasing amount of software which is locked up, either on disk or in a cartridge, discourages not only pirates, but also tinkering with and improving it. Manufacturers, possibly responding to the climate of opinion or possibly causing it, put more effort into tutorials and beginners guides than ever before (commendable) but leave out of the manuals much of the technical information they used to include, or like Apple, charge extra for the real manuals.

There is absolutely nothing wrong with using a computer as an appliance, but I suggest there are good reasons why everyone should learn at least a little about programming.

WHY learn to program?

I am not suggesting everyone should, or can, become an expert programmer. But there are several reasons why everyone should work to gain a nodding acquaintance with simple programming.

First off, because it is fun. It may not seem so at first, but once you have, on your own, forced the computer to do exactly what you told it to, the thrill is immense. The challenge is enormous; so are the rewards. Americans seem more ready to accept this sort of mental challenge than we do. Why?

Secondly, so you can, if you need to, modify other programs. I was once asked if I would modify a program which had been bought in the USA. Naturally, I agreed — the pay offered was good. But the only changes needed were in spelling of words and names of items — timber instead of lumber, aluminium instead aluminum, for example. The program user could have learnt to do that much!

The third reason is so that you can appreciate and criticise programs you meet. After writing a program, no matter how simple, you will understand what is involved in some of the complex, clever, packages you have bought. You will also be able to criticise, and reject programs which aren't good enough, or which make you the slave of the machine rather then the reverse.

The fourth reason: to sharpen your mental processes and help you sort out your ideas. You really do know something if you can program it!

WHO should learn to program?

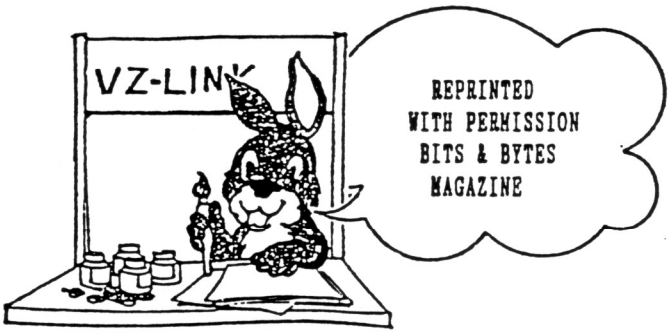
Basically, everyone.

School children — not to get a job, but so they really do understand what computers are like.

Parents — so that they can keep up with their children, and even have some constructive fun with them.

Business people — not to write business software (there isn't time for that) but so that they can decide what software to buy, or have an idea of what their programmers are saying and why they are charging so much. Fewer bad programs would be around if bosses knew just enough to tell a programmer why the work wasn't good enough.

Everyone — to be little more independent.



WHAT to program?

Every beginning programmer should have this motto written in large letters above the keyboard:

KISS: (Keep It Simple Stupid). Anything will do, provided it is simple. Usefulness is also worthwhile. Maybe your first programs won't be all that useful. Stick to things you know, or simple games. Perhaps a simple program can be used for some simple business task. Maybe you can keep a list of people to whom you send Christmas cards. What calculations have you done in the last month?

WHICH programming language?

It is fashionable to knock BASIC. But for most people, BASIC is the only language they have available. This simplifies the choice immensely.

Of course, there are better languages, and decades of teaching schoolchildren has convinced me, against previous prejudice, that a language such as PASCAL might be better first. Certainly, BASIC has to be overcome eventually, but it is easier to learn good habits in PASCAL than BASIC, and the effort of learning a second language is much less than the effort of learning the first.

HOW do I go about it?

Programming is a bit like riding a bicycle. Once you learn, you will wonder what all the fuss was about. The main thing is to get down to the keyboard and get stuck in.

Several sources of help are available. Most important, a friend who knows a little more than you, and can offer a suggestion when things go wrong. Join your nearest computer club — they are always willing to help. Many schools and computer shops run night classes — the school ones are almost always cheaper and often represent a bargain.

Read programs in magazines. Type some in, thinking as you type about the program instructions, and try to see why they are written as they are. Take a program written for a machine different to yours and convert it.

NARRATION FILE CREATOR

INTRODUCTION FOR
NARRATION FILE CREATOR

Picture this:

You're at the keyboard of the trusty VZ 200 working on a game program which requires a considerable amount of narrative. It has been a long hard slog, but you're on the home stretch. Your fingers are getting slower as cramp sets in. Your back is killing you. But by all that's holy, you are GOING TO FINISH IT!!

You turn the page of your rough-scrawled layout. Oh thank goodness!! Only 77 more lines to go! The little line on the bottom of the page seems to grin at you:

65500 END

You type in the next line, hurrying to finish. You want to see if the new game works!

Then, to your unimaginable horror...!

?OUT OF MEMORY ERROR!!!!!

AAAHHHRRRRR!!!! NO!! AFTER ALL THAT WORK!!!!

HOLD ON A MINUTE!! There is a way out of this! Yes indeed! Why not have all the PRINT commands, such as narration, instructions and so on, on a separate set of files on the DOS disc! The relevant part of the program can call up a particular file simply by having it load in and PRINT.

But what about the OPEN/IN#/CLOSE Subprograms? Don't they take up just as much room as the original PRINT command?

Yes, they would, if you used a different subprogram for each PRINT statement.

But why not POKE the reference number into a OPEN/IN#/CLOSE Subprogram right at the beginning of your game program. Follow that with an all-round PRINT Subprogram, and the rest of your PRINT commands are reduced to naming the file reference and a GOSUB!

But how do you make up the files in the first place? Simple! Use my FILE CREATION program, and it's sister, the FILE CHECK program to check the file and see that it says what you want it to say!

The FILE CHECK program may be modified and placed at the start of your game program as the recall for the files.

As you look over the two programs, you will see that there is a completely separate sub-program starting at line 2000 of each program.

This is a screen dump which PEEKS at each symbol of the PRINT part, to check on the file identification POKE values (there are three), then check that the variables of AD are the same. It lists the lines 500 to 504 after running, so that any changes may be made.

For each file I have denoted 15 lines which fills the screen with print nicely, but DIM A$ is 20, so there are five extra spaces. If you wish to put these to other uses, variables, etc, then go ahead: you will have no problems.

But if you change lines 7 to 15 at all, be sure to check the values of AD using 'RUN 2000'.

Each line is 31 letters long. This automatically fits each line into the page neatly. If you try to go over 31 the program will automatically forbid it.

Don't use any commas; the program will not accept them.

To end a file, type in a ';' (semi-colon). This will mark the end and tell the computer to return (line 607).

The identification code is made up from three letters or numbers or both. This gives many thousands of variations, more than enough for the average game program.

And not only games! I'm sure that many of you will be able to come up with your own use for these programs! Go to it!

OK, that's it. All yours, and I hope you find them useful!

Steve Dale

NARRATION FILE CREATOR

```
1 GOTO40
8 OPEN"ABC",1
9 FORZ=1TO16:PR#"ABC",A$(Z):IFA$(Z)=";"THEN10:ELSENEXTZ
10 CLOSE"ABC":RETURN
12 FORPO=0TO2:POKEAD+PO,ASC(MID$(N$,PO+1)):NEXTPO
15 RETURN
30 '**NARRATION FILE CREATOR**
31 'COPYWRITE STEVE DALE 1990
32 'THIS PROGRAM IS TO MAKE USE
33 'OF ALL AVAILABLE MEMORY FOR
34 'VERY LARGE PROGRAMS, USING
35 'THE DOS RECORD FILES FOR
36 'NARRATION OF THE PROGRAM.
40 CLEAR500:DIMA$(20):GOSUB500
50 GOSUB600
60 GOSUB8
100 PRINT"AGAIN";:INPUTT$:IFT$="Y"THENRUN
110 END
500 INPUT"FILE NAME";FILE$:AD=31482:N$=FILE$:GOSUB12
502 AD=31505:GOSUB12
504 AD=31545:GOSUB12
510 RETURN
600 FORZ=1TO15
602 PRINT"INPUT ";Z;:INPUTA$(Z)
605 IFLEN(A$(Z))>31THENPRINT"TOO LONG!":GOTO602
607 IFA$(Z)=";"THENRETURN
610 NEXTZ:LETA$(16)=";":RETURN
700 END
2000 'CHECK ON FILE TITLE
2005 CLS:POKE30777,1:PRINT" SPACE START/STOP  Q=QUIT"
2010 FORL=31470TO31600:PRINTTAB(9)L;PEEK(L);CHR$(PEEK(L))
2020 IFINKEY$=" "THEN2030ELSEIFINKEY$="Q"THEN2040
2030 IFINKEY$=""THEN2020ELSENEXT
2040 PRINT"AGAIN";:INPUTT$:IFT$="Y"THEN2000ELSELIST500-504
```

NARRATION FILE CHECK

```
1 GOTO40
7 OPEN"ABC",0
8 FORZ=1TO16:IN#"ABC",A$(Z)
9 IFA$(Z)=";"THEN11
10 NEXTZ
11 CLOSE"ABC":RETURN
12 FORPO=0TO2:POKEAD+PO,ASC(MID$(N$,PO+1)):NEXTPO
15 RETURN
20 '*********NARRATION FILE CHECK************
21 'THIS PROGRAM IS TO BE USED IN CONJUNCTION WITH THE N.F.C.
22 'COPYWRITE STEVE DALE 1989
40 CLEAR 500:DIMA$(20):GOSUB500:GOSUB7:CLS
50 FORZ=1TO15:IFA$(Z)<>";"THENPRINTA$(Z):NEXTZ
100 PRINT"AGAIN";:INPUTT$:IFT$="Y"THENRUN
110 END
500 INPUT"FILE NAME";FILE$:AD=31482:N$=FILE$:GOSUB12
502 AD=31505:GOSUB12
504 AD=31551:GOSUB12
510 RETURN
2000 'LINE CHECK
2010 CLS:POKE30777,1:PRINT"   SPACE START/STOP Q=QUIT"
2020 FORL=31400TO31600:PRINTTAB(9)L;PEEK(L);CHR$(PEEK(L))
2030 IFINKEY$=" "THEN2040ELSEIFINKEY$="Q"THEN2050
2040 IFINKEY$=""THEN2030ELSENEXT
2050 LIST500-504
2060 END
```

BUY SELL SWAP

This new service is available to ALL members. And you can advertise free.
All you need do is send in your advert in it's final format (as you would like it to appear in the newsletter).

I will then cut and paste to fit the available space.
The advertised item,(buy sell or swap) does not have to be VZ or computer related.

NOTE: This section will only appear as space permits.

WANTED TO BUY OR SWAP

VZ printer interface, will buy or swap for Ram pack.
Contact Peter Hill ph 761-451

MICRODATA

```
0 '++++++++++++++++++++++++++++     "+ MICRODATA2 +   MICRO   +
1 '+++++++++++++++   MAGIC   +                     +++++++++++++
2 '   UPGRADED.....JUNE 1987
100 CLS:CLEAR10000:DIMZ$(402):COLOR3,1:POKE30744,1
110 GOSUB940:PRINT@36,"MICRODATA"
111 FORJ=1TO400STEP10:Z$(J)="*":NEXT
120 PRINT@100,"ENTER FILE NAME":PRINT:PRINT"    ";:INPUTF$
130 IFF$=""THENF$="NO NAME"
140 CLS:PRINT@67,"MENU  ";F$:PRINT
150 PRINT"    CATALOGUE":PRINT"    PAGE":PRINT"    LOAD"
160 PRINT"    SAVE":PRINT"    HARD COPY":PRINT"    NEW FILE"
170 PRINT"    ALPHABETICAL SORT":PRINT"    ENTER INDEX NUMBER"
190 A$=INKEY$:A$=""
200 A$=INKEY$:IFA$=""THEN200
210 IFA$="P"THEN390
220 IFA$="H"THEN520
230 IFA$="A"THEN850
240 IFA$="C"THEN280
250 IFA$="S"THEN1080
260 IFA$="L"THEN1000
270 IFA$="N"THENRUNELSE200
280 CLS:PRINT@3,"CATALOGUE  ";F$:PRINT
290 A=1:FORC=1TO40
292 IFZ$(A)="*"THEN310
294 IFZ$(A)="-"THEN310ELSEPRINT"PAGE"C;
300 PRINTZ$(A):GOSUB330
310 A=A+10
320 NEXT:GOSUB340:GOTO400
330 FORT=1TO400:NEXT:SOUND31,1:RETURN
340 A$=INKEY$:A$=""
350 PRINT"END OF CATALOGUE":PRINT"MENU  PAGE":SOUND28,2
360 A$=INKEY$:IFA$=""THEN360
370 IFA$="M"THEN140
372 IFA$="P"THEN390ELSE360
380 GOTO400
390 CLS:PRINT@230,"PAGE NUMBER";:INPUTP:IFP<0ORP>40THEN390
400 L=P*10-9:J=P*10-9:CLS
410 PRINT"PAGE "P;"  ";F$:PRINT
420 FORI=1TO10:PRINTUSING"###";L;:PRINT" ";Z$(J):L=L+1:J=J+1
430 NEXT:SOUND25,1
440 PRINT" ENTER LINE INSERT":PRINT" ALPHA DELETE PAGE MENU"
450 A$=INKEY$:A$=""
460 A$=INKEY$:IFA$=""THEN460
470 IFA$="L"THEN690
480 IFA$="I"THEN720
490 IFA$="D"THEN780
500 IFA$="P"THEN390
502 IFA$="A"THEN850
510 IFA$="M"THEN140ELSE460
520 PRINT@418,"COPY TO PRINTER"
530 A$=INKEY$:A$=""
540 PRINT@450,"ENTER PAGE(S)  FILE  MENU";:SOUND28,1
550 A$=INKEY$:IFA$=""THEN550
560 IFA$="F"THENP=1:F=40:GOTO610
570 IFA$="M"THEN140
580 IFA$="P"THEN590ELSE550
590 GOSUB990:PRINT@417,"ENTER START PAGE NUMBER";:INPUTP
600 PRINT@455,"FINAL PAGE NUMBER";:INPUTF:SOUND28,1
```

```
610 X=1:GOSUB990:LPRINTF$:PRINT
620 FORI=PTOF:LPRINT"PAGE"I
630 FORJ=XTOX+9:IFZ$(J)="-"THEN650
640 LPRINTJ;Z$(J)
650 NEXT
660 X=X+10
670 NEXT
680 LPRINT:GOTO190
690 PRINT@391,"LINE NUMBER";:INPUTL
700 FORD=417TO447:PRINT@D,CHR$(32):NEXT:SOUND30,1
710 PRINT@417,"NEXT";:INPUTZ$(L):GOTO400
720 PRINT@391,"LINE NUMBER";:INPUTL
730 FORD=417TO447:PRINT@D,CHR$(32):NEXT
740 PRINT@418,"NEXT";:INPUTN$
750 FORJ=400TOLSTEP-1:Z$(J+1)=Z$(J):NEXT
760 Z$(L)=N$
770 GOSUB1600:GOTO400
780 PRINT@391,"LINE NUMBER";:INPUTL
790 X=400-L
800 FORJ=LTOX:Z$(J)=Z$(J+1)
810 Z$(J)=Z$(J+1)
820 IFZ$(J)="*"ORZ$(J)="-"THEN840ELSENEXT
830 Z$(400)="-"
840 GOSUB1500:GOTO400
850 PRINT@449,"SORT START LINE";:INPUTS:SOUND15,1
860 PRINT@449,"SORT FINAL LINE";:INPUTF:J=0:I=0
870 PRINT@416,"[illegible]":SOUND19,3
880 IFS<1ORS>400ORF<SORF>400THEN970
890 FORJ=STOF:IFZ$(J)="-"THENZ$(J)="  ":NEXT
900 S=S+1:K=F-S:FORJ=1TOK+1
910 FORI=FTOSSTEP-1
920 IFZ$(I)<Z$(I-1)THENTP$=Z$(I):Z$(I)=Z$(I-1):Z$(I-1)=TP$
930 NEXT:NEXT:GOTO400
940 FORJ=1TO400:Z$(J)="-":NEXT:RETURN
950 FORD=417TO449:PRINT@D,CHR$(32):NEXT:SOUND25,1:RETURN
960 CLS:PRINT@262,"[illegible]":RETURN
970 PRINT@416,"[illegible]":SOUND4,9
980 GOTO400
990 FORA=417TO478:PRINT@A,CHR$(32):NEXT:SOUND30,1:RETURN
1000 PRINT@417,"[illegible]":COLOR4
1010 PRINT" [illegible]":SOUND28,9:COLOR5
1020 INPUT#"",F$:PRINTF$:PRINT
1030 FORI=1TO400
1040 INPUT#"",Z$(I)
1045 IFZ$(I)="*"THEN1060
1048 PRINTI;Z$(I)
1050 NEXT
1060 PRINT" [illegible]":SOUND30,9:GOTO140
1070 C$="*":GOSUB990
1080 PRINT@417,"[illegible]"
1090 A$=INKEY$:A$=""
1100 PRINT@449,"ENTER  FULL FILE OR PART ONLY ":SOUND28,1
1110 A$=INKEY$:IFA$=""THEN1110
1120 IFA$="F"THENGOSUB990:S=1:F=400:GOTO1180
1130 IFA$="P"THEN1150
1140 IFA$="M"THEN140ELSE1110
1150 GOSUB990:PRINT@417,"ENTER START PAGE NUMBER";:INPUTP
1160 PRINT@455,"FINAL PAGE NUMBER";:INPUTF:GOSUB990
1170 S=P*10-9:F=F*10:GOSUB950
1180 PRINT@417,"[illegible]":SOUND28,9
1190 PRINT#"",F$:PRINT@425,"RECORDING":PRINT@449,F$
1200 FORI=STOF
```

```
1210 PRINT#"",Z$(I)
1212 IFZ$(I)="*"THEN1230
1214 PRINTI;Z$(I)
1220 NEXT
1230 PRINT" [illegible]":SOUND30,9:GOTO140
1500 G=INT(L/10)*10+10:IFZ$(G)="*"THENZ$(G)="-"
1505 IFZ$(G-9)="-"THENZ$(G-9)="*"
1510 FORI=GTO400STEP10:IFZ$(I)="*"THENZ$(I)="-"
1520 NEXT:G=G+1
1530 FORI=GTO401STEP10:IFZ$(I)="-"THENZ$(I)="*"
1540 NEXT
1550 RETURN
1600 G=INT(L/10)*10+11
1610 FORI=GTO400STEP10:IFZ$(I)="-"THENZ$(I)="*"
1620 NEXT:G=G+1
1630 FORI=GTO402STEP10:IFZ$(I)="*"THENZ$(I)="-"
1640 NEXT
1650 RETURN
2000 GOTO140
```

HI SCORE REGISTER

BY PETER HILL.

This Subroutine is designed to enable you to save your High score to Disk.
All you need to do is add this rather long Subroutine to your program. At the end of your game.

```
1000 H=10:GOTO 60015
```

LINE 1000 is an example of the last line of your game which will send you to the Score Register.
The Variable "H" should be used to hold your score.

NOTE: Before playing, you will need to construct a Data file on your disk. You can use the following lines to do this.
Once done you can erase these lines.

```
0 NH$="":L=5
1 OPEN"H/SCORE",1
2 PR#"H/SCORE",NH$,L
3 CLOSE"H/SCORE"
```

LINE 59995 is where the subroutine goes after finishing. You would return to start a new game from here. Remember to reset the score to ZERO before you do. H=0.

The Subroutine allows you to insert your 3 initials.
It then runs through the Subroutine as follows:-

LINE 60160
Print your score.

LINE 60165 - 60175
Get high score from disk.

LINE 60190
Print high score.

LINE 60195
Compare high score with your score.

LINE 60200
Print new high score, if your score is higher.

LINE 60205
If your score is higher, replace old high score with your high score.

LINE 60210 - 60220
Save score and initials to disk.

LINE 60225
Return to Line 59995

This subroutine uses the following Variables:-
H, C, P, S, L, I, HH, NH

```
1000 H=10:GOTO60015
59995 SOUND31,2:END
60000 'LINE 1000 IS AN EXAMPLE  OF THE LINE THAT WILL DIRECT
60005 'THE PROGRAM TO THE HIGH- SCORE ROUTINE, THE VARIABLE "H"
60010 'IS THE PRESENT HIGH SCORE AT THE END OF GAME
60015 CLS
60020 HH$="":C=0:PRINT
60025 PRINT"      ***SCORE REGISTER***  ":PRINT
60030 PRINT" USE THE ARROW KEYS*****"
60035 PRINT" TO MOVE THE ARROW UNDERNEATH"
60040 PRINT" THE LETTERS OF YOUR INITIALS.":PRINT
60045 PRINT"        *** PRESS SPACE ***"
60050 PRINT" TO ENTER THE LETTERS IN."
60055 PRINT@451,"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
60060 P=483:S=5
60065 PRINT@P,"^";
60070 K$=INKEY$:I$=INKEY$
60075 IFI$="M"THEN60095
60080 IFI$=","THEN60110
60085 IFI$=" "THEN60125
60090 GOTO60065
60095 IFP=483THEN60070
60100 P=P-1:S=S-1:PRINT@P+1," ";:SOUNDS,1
60105 GOTO60065
60110 IFP=508THEN60065
60115 P=P+1:S=S+1:PRINT@P-1," ";:SOUNDS,1
60120 GOTO60065
60125 PRINT@P-1,"<";:PRINT@P+1,">";
60130 I$=CHR$(P-418)
60135 HH$=HH$+I$+"."
60140 PRINT@320,HH$
60145 PRINT@P-1," ";:PRINT@P+1," ";
60150 C=C+1:FORW=0TO200:NEXT:IFC=3GOTO60160
60155 FORG=1TO200:NEXT:GOTO60065
60160 GOSUB60230:CLS:PRINT:PRINT"YOUR SCORE  ";HH$;H
60165 OPEN"H/SCORE",0
60170 IN#"H/SCORE",NH$,L
60175 CLOSE"H/SCORE"
60180 GOSUB60230
60185 ERA"H/SCORE"
60190 PRINT"HIGH SCORE  ";NH$;L
60195 IFH>LGOTO60200ELSE60210
60200 PRINT"NEW HIGH SCORE  ";HH$;H:GOSUB60230
60205 NH$=HH$:L=H
60210 OPEN"H/SCORE",1
60215 PR#"H/SCORE",NH$,L
60220 CLOSE"H/SCORE"
60225 GOTO59995
60230 FORW=0TO700:NEXT:RETURN
```

DRIVE HEAD PARK ROUTINE
WRITTEN BY D.MITCHELL

HANDS UP THOSE THAT HATE THE NOISES WHICH COME FROM THE DRIVE WHEN THE VZ IS RESET OR POWERED UP. THE REASON FOR THIS NOISE IS THE DOS DOESN'T KNOW WHERE THE DRIVE HEAD WAS LEFT SO IT MUST BE SET TO TRACK ZERO HENCE DOS MOVES THE DRIVE HEAD 40 TIMES.

THIS LITTLE ROUTINE WHICH I CALLED PARK, IS USED IN THE LATEST VERSION WORD PROCESSOR PATCH. IT PLACES THE DRIVE HEAD OF DRIVE 1 AT TRACK 39.

I USE THIS ROUTINE BEFORE I TURN THE POWER OFF, ON THE VZ AND WHEN I SWITCH ON ALL THAT I HEAR IS THE DRIVE HEAD MOVING TO TRACK ZERO WITHOUT THE BASHING THAT WENT ON BEFORE.

THE ROUTINE IS SUPPLIED IN ASSEMBLER FOR THOSE THAT WAY INCLINED OR IN BASIC FOR THE OTHERS.

LISTING FOR BASIC VERSION

```
10 FOR I=29184 TO 29261: READ A :POKEI,A: B=B+A : NEXT
20 IF B <> 9491 THEN PRINT "ERROR IN DATA": END
30 PRINT "INSERT DISK,CLOSE DOOR & PRESS  RETURN"
40 A$=INKEY$:IF INKEY$ <> CHR$( 13 ) THEN 40
50 IF PEEK( 16384 ) = 170 THEN 70
60 PRINT "NO DISK DRIVE TO USE" : END
70 BSAVE"PARK",7200,724D
80 END
100 DATA243,205,8,64,62,39,253,150,20,40,4,71,205,22,114,
205,11
110 DATA64,251,195,25,26,253,54,20,39,203,32,253,126,56,79,7
120 DATA245,253,119,56,177,205,63,114,205,54,114,241,205,
63,114
130 DATA205,54,114,16,231,201,197,1,2,0,205,56,64,193,201,230
140 DATA15,79,62,240,253,166,51,177,253,119,51,211,16,201
```

THE ASSEMBLER CODE

```
001 ;DISK HEAD PARK ROUTINE
002 ; ORIGIN 7200 HEX
003        DI
004        CALL 4008H
005        LD   A,27H
006        SUB  (IY+14H)
007        JR   Z,END
008        LD   B,A
009        CALL SIN
010 END    CALL 400BH
011        EI
012        JP    1A19H
013 SIN    LD    (IY+14H),27H
014        SLA   B
015 SIN1   LD   A,(IY+38H)
016        LD    C,A
017        RLCA
018        PUSH   AF
019        LD     (IY+38H),A
020        OR     C
```

DRIVE PARK ROUTINE CONT. . . .

```
021        CALL   MOVE
022        CALL   DLY
023        POP    AF
024        CALL   MOVE
025        DJNZ   SIN1
026        CALL   DLY
027        RET
028 DLY    PUSH   BC
029        LD     BC,2
030        CALL   4038H
031        POP    BC
032        RET
033 MOVE   AND    0FH
034        LD     C,A
035        LD     A,0F0H
036        AND    (IY+33H)
037        OR     C
038        LD     (IY+33H),A
039        OUT    (10H),A
040        RET
```

THIS TIME I AM NOT GOING TO GIVE YOU A BLOW BY BLOW DISCRIPTION, IF YOU ARE INTERESTED IN WHAT IS HAPPENING, YOU CAN LOOK UP THE TECHNICAL MANUAL FOR THE DOS ENTRY POINTS.

ALL THAT WE ARE REALLY DOING IS FINDING THE DIFFERENCE BETWEEN THE CURRENT LOCATION OF THE DRIVE HEAD AND TRACK 39 AND MOVING THE HEAD TO TRACK 39. SO WHEN THE VZ IS RESET OR POWERED UP THE DRIVE HEAD MOVES TO TRACK ZERO.

HINTS & TIPS FOR DISK DRIVE USERS

THE BACK OF MOST DISK JACKETS CONTAIN USEFUL WARNINGS ON AVOIDING PROBLEMS WHICH CAN DAMAGE DISKS OR INFORMATION STORED ON THEM. THERE ARE OTHERS WHICH YOU SHOULD BE AWARE OF :-

DISK LABELS - THE ONES SUPPLIED WITH THE DISKS USUALLY STICK LIKE CEMENT AND ARE HARD TO REMOVE WITHOUT DAMAGING DISK. THE SOLUTION IS SIMPLE, USE REMOVABLE LABELS AVAILABLE FROM NEWSAGENTS OR STATIONERS IN VARIETY OF SHAPES AND SIZES.

MARKING LABELS - DO NOT USE BIROS, PENCILS OR ANY HARD POINTED WRITING IMPLEMENT. THE DISK ITSELF IS FAIRLY SOFT AND IS EASYLY DAMAGED BY PRESSURE. FELT TIP PENS ARE IDEAL FOR THE JOB.

HINTS & TIPS FOR TAPE USERS . . .

TAPE LABELS - LIKE DISK LABELS THEY ALSO ARE HARD TO REMOVE SO MANY USERS JUST STICK ONE OR MORE LABEL ON TOP OF ANOTHER WHICH CAN CAUSE A PROBLEM. TOO MANY LABELS ON ONE SIDE CAN ACTUALLY LIFT THE CASSETTE UP SLIGHTLY CAUSING MISALIGNEMENT WITH THE HEAD. THE CURE IS SIMPLE, REMOVE OFFENDING LABELS AND REPLACE WITH REMOVEABLE LABELS WHICH ARE AVAILABLE IN A VARIETY OF SIZES.

SELDOM USED TAPES - IF YOUR SELDOM USED PROGRAM TAPE REFUSES TO LOAD PROPERLY THEN TRY FAST FORWARD TO END & REWIND TO BEGINNING AGAIN. WHAT HAPPENS IS IF A TAPE IS NOT USED FOR A LONG TIME THE LAYERS OF TAPE TEND TO STICK TOGETHER A BIT AFFECTING PROPER LOADING. FAST FORWARD & REWIND USUALLY CURES PROBLEM.

NEWSLETTER INDEX
ISSUES 34 TO 40

ISSUE 34..

PROGRAM-JUKE BOX (KEN CLARKE)
VZ DOS PART 2
PROGRAM-DISK UTILITY (JOE LEON)
PROGRAM-SUPER PREDICTIONS
NEWSLETTER INDEX-ISSUES 21 to 26

ISSUE 35..

PROGRAM-TRIGONOMETRY (PAUL SHEPPARD)
PAGED DISK MENU (P.& J.LEON)
HI RES SCREEN MOVE (BOB KITCH)
MAKING C.W.P WORK (JOE LEON)
VZ DOS PART 3
NEWSLETTER INDEX-ISSUE 27 TO 33

ISSUE 36..

PROGRAM-ALFA TIMER (STEVEN DILLON)
INSIDE THE VZ
VEEZED TEXT GRID
PROGRAM-LOTTO (ALAN TOZER)
MICRO MAGIC PART 1 (BUTCHER)

ISSUE 37..

MICRO MAGIC PART 2 (PRINTER EXPERIMENT)
PROGRAM-BUTCHER (FULL LISTING)
VZ DOS PART 4
4k - 64k RAM/EPROM (JOE LEON)
CONVERSION CHART 1
DISK DRIVE ENQUIRY-A PLEA FOR HELP FROM LOU ELLMERS

ISSUE 38..

MICRO MAGIC PART 3 (MICRO TYPE)
DISK DRIVE MAINTENANCE (FROM VZ USER MAGAZINE)
PROGRAM-MORSE TUTOR (STEVE DALE)
CONVERSION CHART 2

ISSUE 39..

MICRO MAGIC PART 4 (MICRO TYPE CONTINUED)
DISK DRIVE ENQUIRY, AN ANSWER (L WHETTON)
SOFTWARE REVIEW- WHIZKID SPYCATCHER (BRENDA HILL)
CSAVE/CLOAD (DAVE MITCHELL)
PROGRAM-MOTOR SPORT (PART 1)
CONVERSION CHART 3
NOTES & ERRATA-CORRECTION TO MORSE TUTOR

ISSUE 40..

PROGRAM-MOTOR SPORT (PART 2) (M BAXTER)
GETTING TO GRIPS WITH YOUR JOYSTICKS (DAVID LAMBOURN)
PROGRAM-ST GEORGE & THE DRAGON (KEN CLARKE)
MICRO MAGIC PART 5 (RATTLESNAKE)